Programmation Python

De la base aux concepts avancés de Python

Audrey Kouessi FANGNON & Johannès HOUNSINOU

Bénin Excellence - Fondation Vallet / FSS COTONOU

30 novembre 2024

Plan

1. Entrainement
2. Cryptographie
3. JEU : Pierre-Feuille-Ciseaux

Entraînement Level = Beginner

Exercice 1 - Fonction sans paramètre

Écrivez la Fonction repete_mon_nom() qui demande le nom de l'utilisateur et qui affiche 5 fois ce nom en faisant à chaque fois un retour à la ligne.

Exercice 2 - Fonction avec paramètre

Écrire une fonction deux_egaux(a, b, c) qui reçoit trois nombres en paramètre et qui renvoie la valeur booléenne True si au moins deux de ces nombres ont la même valeur, et la valeur booléenne False sinon.

Exercice 3 - Fonction avec paramètre par défaut

Écrivez une fonction somme(a, b) qui retourne la somme de deux valeurs entières a et b. Par défaut, la valeur de a est 0 et la valeur de b est 1.

Entraînement — Level = Intermediate

Exercice 4 - Table de Multiplication

Écrivez une fonction table_multiplication(n) qui reçoit un entier naturel n non nul et qui affiche la table de multiplication de n et renvois None si $n \notin \mathbb{N}^*$

Exercice 5 - Les nombres de Catalan

Les dix premiers nombres de Catalan (pour n de 0 à 9) sont : 1 - 1 - 2 - 5 - 14 - 42 - 132 - 429 - 1430 - 4862. Le n-ième nombre de Catalan, est défini par :

$$C_n = \frac{(2n)!}{(n+1)!n!}$$

Écrire une fonction catalan(n), où n est un nombre entier positif ou nul, qui renvoie la valeur du n-ième nombre de Catalan en utilisant une autre fonction factoriel(n) qui calcule le factoriel.

Cryptographie Level = Advanced

Le codage de Césare.

Le codage de Césare est une manière de crypter un message de manière simple.
On choisit un nombre (appelé clé de codage) et on décale toutes les lettres de notre message du nombre choisi.

Exemple

- Si on choisit comme clé le nombre 2, alors : la lettre A deviendra C, la lettre B deviendra D ... et le Z deviendra B.
- Pour decoder, il suffit de faire la même chose avec l'opposé de la clé.

Cryptographie Level = Advanced

Consignes

Ecrivez deux fonctions coder(secret, cle) et decoder(secret, cle) qui prennent une chaine de caractère et un entier naturel compris entre 0 et 26 (y compris les bornes) puis retourne respectivement la chaine de cractère codée et celle décodée suivant le codage de Césare.
Pour cella, on écrira deux versions des fonctions.

- La version v1 fonctionne uniquement sur une chaine de caratère composée de lettre majuscule.
- La version v2 plus robuste fonctionne sur tout type de chaine de caratère.

Cryptographie - Ressource Level = Advanced

Consignes (suite)

- Le message à coder ne doit pas contenir de caractère accentué.
- Les ponctuations et espaces ne doivent pas changer.

Fonction ord et chr

Consulter l'aide sur les fonction ord(), chr()

Code ASCII

(Voir le document associé)

Pierre-Feuille-Ciseaux Level = Advanced

Déroulement

Pierre-feuille-ciseaux est un jeu effectué avec les mains et qui oppose deux joueurs. Ces derniers choisissent simultanement trois coups possible :

- « PIERRE » : Poing fermé
- « FEUILLE » : main ouverte et doigts collés.
- « CISEAUX » : main avec seulement index et majeur en V

La PIERRE bat le CISEAUX (en les émoussant), les CISEAUX battent la FEUILLE (en la coupant) et la FEUILLE bat la PIERRE (en l'enveloppant).

Pierre-Feuille-Ciseaux Level = Advanced

Consigne

Ecrivez un programme qui réalise cinq manches du jeu Pierre-feuille-ciseau. Chaque manche consiste à :

- générer un entier (pseudo) aléatoire entre 0 et 2 compris en utilisant randint.
- lire une des valeurs 0, 1 ou 2 correspondant au coup du joueur.
- afficher le résultat sur une des formes:
  - coup_o bat coup_j : points
  - coup_o est battu par coup_j : points
  - coup_o annule coup_j : points

  Avec : coup_0 -> le coup de l'ordinateur et coup_j -> le coup du joueur
- A la fin des cinq manches, le programme affiche : Perdu, Nul ou Gagné compte tenu du score final.

Pierre-Feuille-Ciseaux Level = Advanced

Début du programme

```
1  import random
2  PIERRE = 0
3  FEUILLE = 1
4  CISEAUX = 2
5
6  s = int(input())
7  random.seed(s)     #Initialisation du générateur
```

Ressource

```
1  import random
2  random.seed(65)   #Initialisation avec 65
3  for i in range(5):
4      print(random.randint(0, 2), end=" ")
```

Pierre-Feuille-Ciseaux Level = Advanced

Ressource

Ce code affiche : 1 1 1 2 0

Exemple 1 - Score négative : Partie perdante

L'exécution du programme avec la suite de nombre : 65 - 0 - 1 - 2 - 1 - 0 doit afficher :

- Feuille bat Pierre : -1
- Feuille annule Feuille : -1
- Feuille est battu par Ciseaux : 0
- Ciseaux bat Feuille : -1
- Pierre annule Pierre : -1
- Perdu

Pierre-Feuille-Ciseaux Level = Advanced

Exemple 2 - Score nul : Partie nulle

L'exécution du programme avec la suite de nombre :
1515 - 0 - 1 - 2 - 1 - 0 doit afficher :

- Ciseaux est battu par Pierre : 1
- Feuille annule Feuille : 1
- Pierre bat Ciseaux : 0
- Ciseaux bat Feuille : -1
- Ciseaux est battu par Pierre : 0
- Nul

Pierre-Feuille-Ciseaux Level = Advanced

Exemple 3 - Score positive : Partie gagnante

L'exécution du programme avec la suite de nombre :
2001 - 0 - 1 - 2 - 1 - 0 doit afficher :

- Ciseaux est battu par Pierre : 1
- Pierre est battu par Feuille : 2
- Feuille est battu Ciseaux : 3
- Pierre est battu par Feuille : 4
- Pierre annule Pierre : 4
- Gagné

Indication (Non obligatoire)

On peut écrire une fonction bat(joueur_1, joueur_2) avec joueur_1, joueur_2 $\in \{0, 1, 2\}$ qui retourne:

Pierre-Feuille-Ciseaux Level = Advanced

Indication (suite)

- -1 si joueur_1 bat joueur_2 ;
- 1 si joueur_1 est battu par joueur_2 ;
- 0 si joueur_1 et joueur_2 font match nul.